Study of equations of Chaotic Systems and their Applications in Image Encryption

Team members:

Jessica John Britto (20PH20014)

Priti Mandal (20PH20029)

Shreya Bhatt (20PH20037)

Order and Chaos - Mini Project

Date: 10-04-2024

1. Introduction

1.1. Image Encryption - General System

Image encryption involves applying cryptographic techniques to transform digital images into unreadable or unintelligible formats, thereby protecting their confidentiality and integrity. This process typically utilises algorithms and keys to scramble the pixel values of an image, rendering it incomprehensible without the appropriate decryption key.

Symmetric and asymmetric encryption, chaos-based methods, transform domain techniques, and visual cryptography are among the common approaches used for image encryption. By securing visual data through encryption, sensitive information can be safeguarded from unauthorised access, tampering, and interception during transmission or storage.

1.2. Image Encryption System using Chaos

The following steps are followed in an Image encryption system utilising chaos for pixel scrambling. Let us assume that Alice wants to share an image with Bob through an open channel which can be eavesdropped.

1. Alice and Bob decide on a pre-shared key which includes a chaotic model (Eg Lorenz model), its parameters (eg a,b,c) and a set of initial points and timestep (x,y,z,dt).
2. Alice wants to share an image of size MxN pixels. Therefore she uses the given Key and a numerical ODE solving method (e.g. RK4) to generate a particle trajectory for MxN steps.
3. Therefore Alice now has an MxN sized pseudorandom key. She calculates the XOR of the pseudorandom key with the original image to get the encrypted image.
4. Alice shares the encrypted image over the open channel.
5. Bob receives the encrypted image.
6. Bob calculated the pseudorandom key using the same pre-shared key (x,y,z,dt) and Lorenz model parameters using the RK4 method.
7. Bob calculates the XOR of the pseudorandom key and encrypted image to get the decrypted image.

1.3. Significance of Chaos in Image Encryption

Chaotic systems are employed in image encryption for their inherent complexity, which can be harnessed to generate highly randomised sequences of numbers. This randomness is vital for creating encryption keys and scrambling image data effectively.

Some applications of chaos in image encryption are:

1. **Pseudorandom Key Generation**: Chaotic systems, such as the Lorenz system or logistic map, can generate sequences of numbers that appear random but are deterministic. These sequences can be used as encryption keys in symmetric encryption algorithms. By utilising chaotic dynamics, the generated keys can exhibit

properties like sensitivity to initial conditions and unpredictability, making them suitable for cryptographic purposes.

2. **Image Scrambling**: Chaotic maps can be used to shuffle the pixels of an image in a nonlinear and unpredictable manner.
3. **Confusion and Diffusion**: Chaotic systems contribute to both confusion and diffusion, which are two fundamental principles in image encryption. Confusion refers to the complexity and randomness introduced into the encrypted data, making it difficult to discern any patterns or structure. Diffusion involves spreading the influence of individual pixels across the entire image, ensuring that changes in one part of the image affect the entire image.

2. Study of Different Models used

2.1. Rossler Attractor

The Rossler attractor has the following set of equations.

$$\frac{dx}{dt} = -y - z$$
$$\frac{dy}{dt} = x + ay$$
$$\frac{dz}{dt} = b + z(x - c)$$

Here are the general fixed points of the system below.

$$x = \frac{c \pm \sqrt{c^2 - 4ab}}{2}$$
$$y = -\left(\frac{c \pm \sqrt{c^2 - 4ab}}{2a}\right)$$
$$z = \frac{c \pm \sqrt{c^2 - 4ab}}{2a}$$

In the x-y plane, when we set z=0, the fixed point we get is a saddle point, which is responsible for the unstable influence in this plane and results in the unstable spiralling. When we consider the z-behaviour, so long as the x>c is valid, the z increases and goes out of the x-y plane towards the z-direction. When x<c, then the z decreases and returns to the x-y plane.

Assume $a > 0$ and z and $\frac{dz}{dt}$ are small. Now studying the system in the $x - y$ plane, system is approximated by

$$\frac{dx}{dt} = -y, \ \frac{dy}{dt} = x + ay$$
$$\frac{d^2x}{dt^2} = -\frac{dy}{dt} = -x + a\frac{dx}{dt}$$

This gives a negatively damped oscillator.

$$\frac{d^2x}{dt^2} + x - a\frac{dx}{dt} = 0$$

The trajectories spiral out of the origin as shown in the figure below which is done for $a = 0.32, \ b = 0.3, \ c = 5.8$. As you could notice here, the spreading of the trajectories is

confined. And, this is possible due to the term $(x - c)$ in $\frac{dz}{dt}$. Therefore, for small b, if $x > c$, the first derivative in z is positive, else it is negative.

Also, these diagrams resemble the Möbius strip.

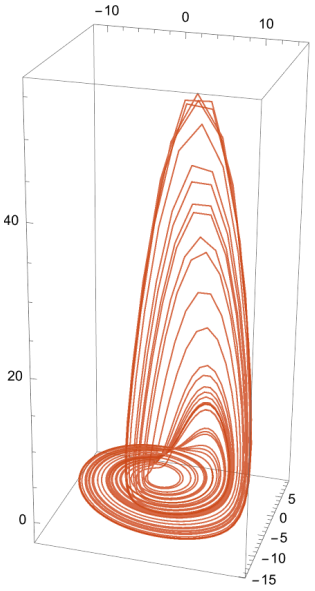
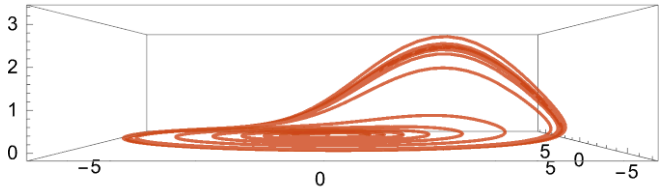
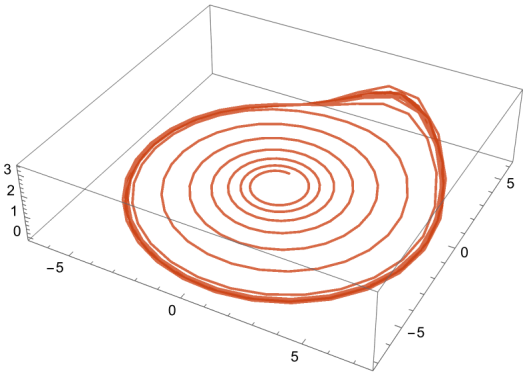
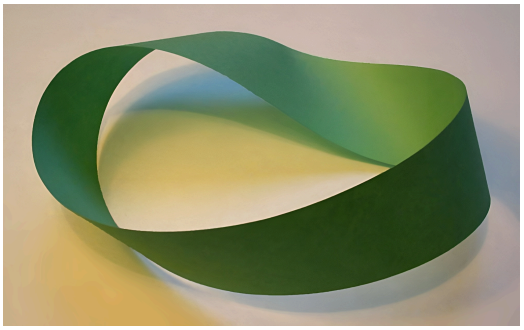


Figure: (a) Phase Portrait plot of the Rossler Attractor for $a = 0.32,\ b = 0.3,\ c = 5.8$ using Mathematica





(a)



(c)

Figure: Phase Portrait plot of the Rossler Attractor for $a = 0.1,\ b = 0.1,\ c = 4$ using Mathematica (a) Front View (b) Default View (c) Möbius strip

We observe sharp lines on these curves, and this plot seems to be in the transient state from current observations. Therefore, we considered a much smaller time step size of 0.0001 and also incorporated the RK4 method as a numerical tool to improve the results. The result of these changes can be found in the plot below.

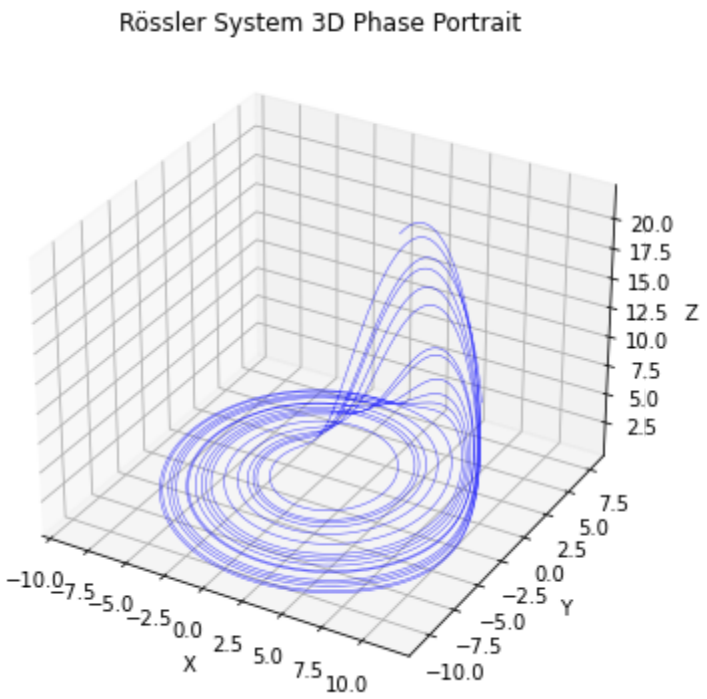


Figure: Phase Portrait plot of the Rossler Attractor for $a = 0.2,\ b = 0.2,\ c = 5.7$

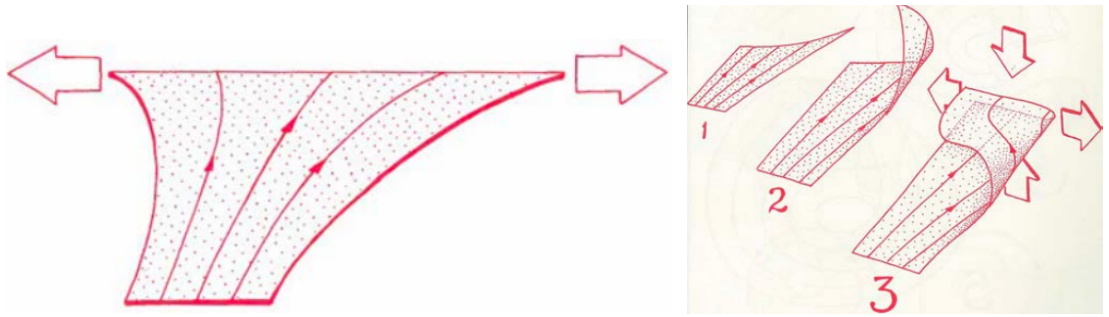
As we can observe the followings from the plot above,

The trajectories diverge in plane by diverging out (stretching). This happens since the fixed point in the centre of the plane is unstable, which influences the trajectories to move out of the plane.

Trajectories leave the $x - y$ plane when $x > c$.

They return to the plane by getting folded, and thereby coming back to the centre of the spiral.

The trajectories behave in the following way - Divergence from the origin creates $x > c$, this results in the increase in the value of z, and x increases. And, this is why, the trajectories spiral out of the fixed point in the centre and also out of the $x - y$ plane. Then x decreases such that $x < c$ for which z decreases and is back in the $x - y$ plane. And, this process repeats. Also, note that here, the trajectories never close as a surface but as a filo dough.



From the bifurcation diagram for this system, we can observe the chaotic behaviour similar to that of the Lorenz attractor.

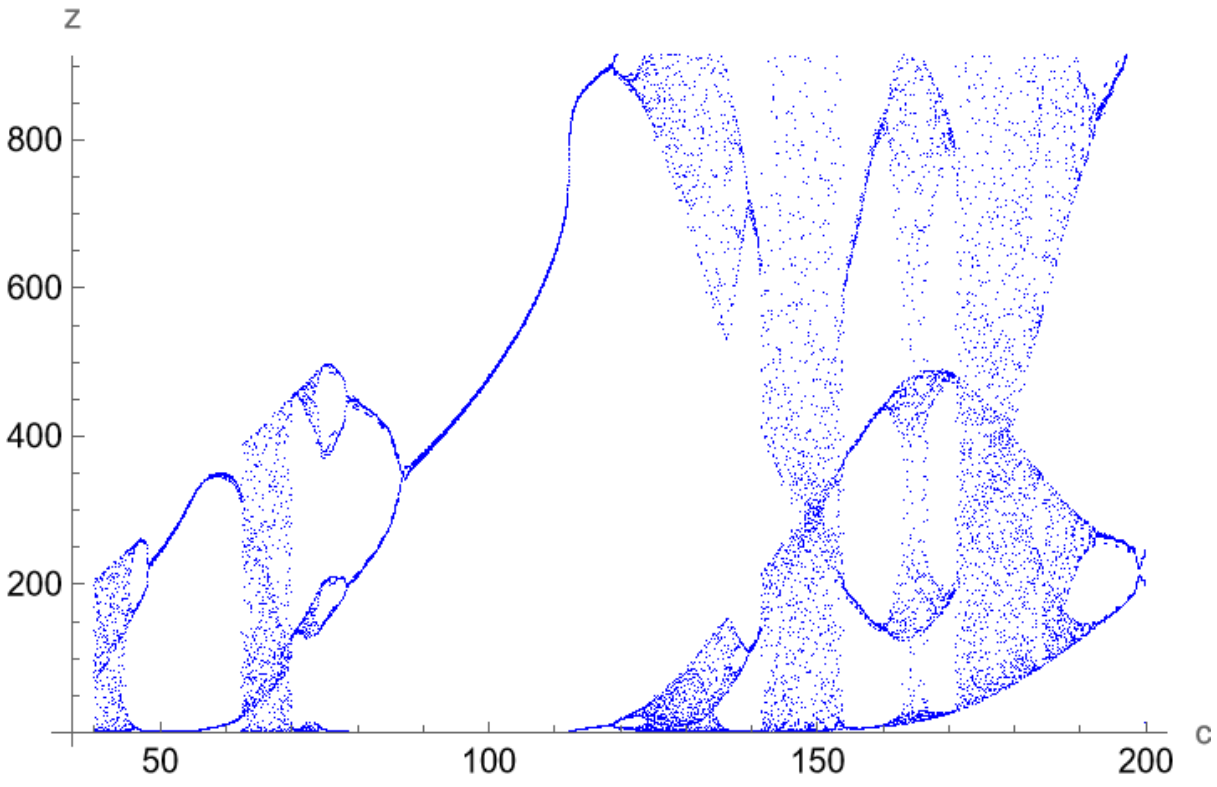


Figure: Bifurcation Diagram of the Rossler Attractor for $a = 0.1$, $b = 0.1$ using Mathematica

2.2. Lorenz Attractor

This system has been studied extensively in coursework for order and chaos too. Therefore, we did not delve into the details of the system and have included detailed phase portraits at the end of the report.

The system of equations for the Lorenz attractor are given by,

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz \end{aligned}$$

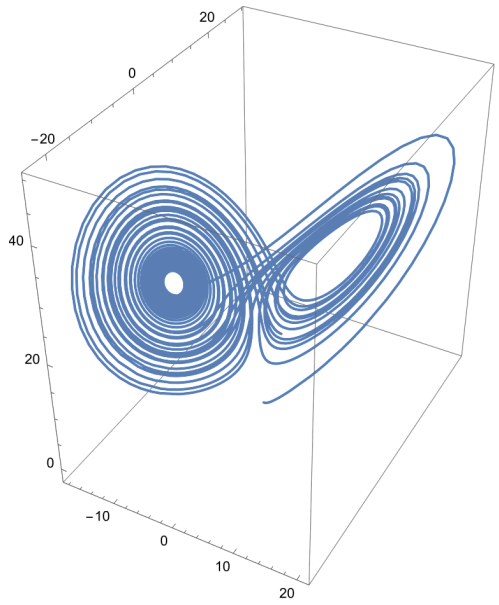


Figure: Phase Portrait plot of the Lorenz Attractor for σ=10, r =28, b =8/3 using Mathematica

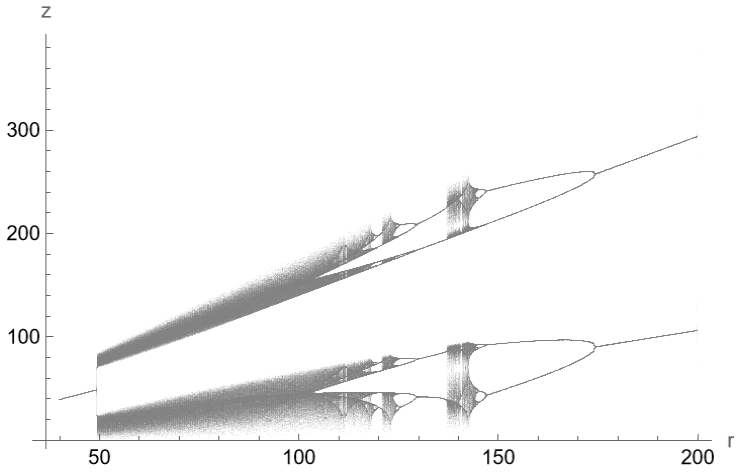


Figure: Bifurcation Diagram of the Lorenz Attractor for σ=10, b =8/3 using Mathematica

2.3. Langford Attractor

The system of equations for the langford attractor are given by,

$$\dot{x} = (z - b)\, x - d\, y,$$
$$\dot{y} = d\, x + (z - b)\, y,$$
$$\dot{z} = c + a\, z - z^3/3 - \left(x^2 + y^2\right)(1 + e\, z) + 0.1\, z x^3.$$

And, the parameter values and the initial conditions we have chosen are as follows

$$a = 0.95, \quad b = 0.7, \quad c = 0.6, \quad d = 3.5, \quad e = 0.25$$

$$x_0 = 0.1, \quad y_0 = 1, \quad z_0 = 0.$$

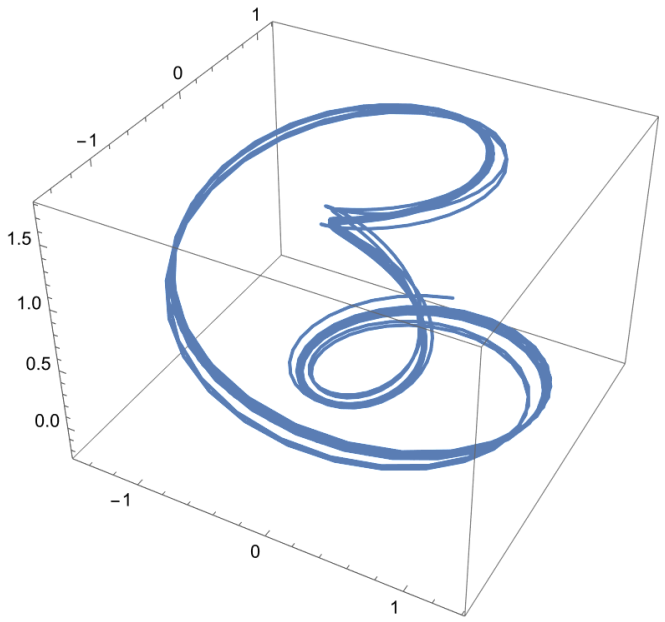


Figure: Phase Portrait plot of the Langford Attractor for the initial conditions and the parameter values mentioned earlier using Mathematica

3. Comparison of Encryption Systems using different models

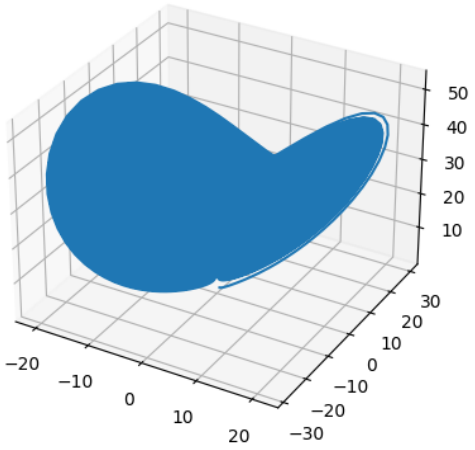
The codebase for this project is at - https://github.com/shreya-bhatt27/encryption-with-chaos

3.1. Lorenz Model Example Images

Parameters used: (10, 28, 2.667)
Key used: (0.1, 1.0, 0.01, 0.01)

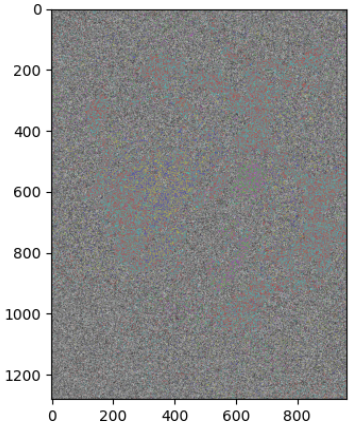
Obtained Images (Image captured by us):

Trajectory generated: Trajectory was generated for 960*1280 time steps of 0.01 dt hence it is filled.



Images obtained
Image size - 960 x 1280 pixels

Original Image

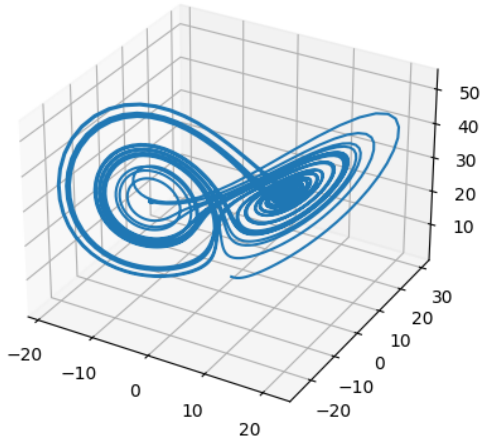


Encrypted Image

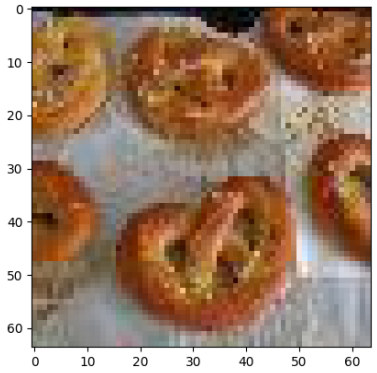
Decrypted Image

Obtained Images (from standard dataset - tiny-imagenet-200):

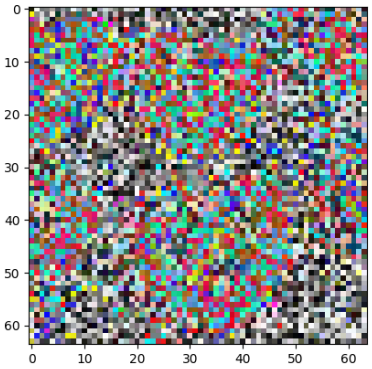
Trajectory generated:



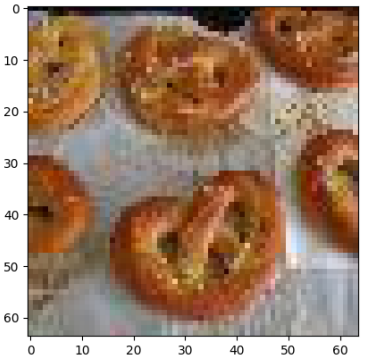
Images obtained:



Original Image



Encrypted Image



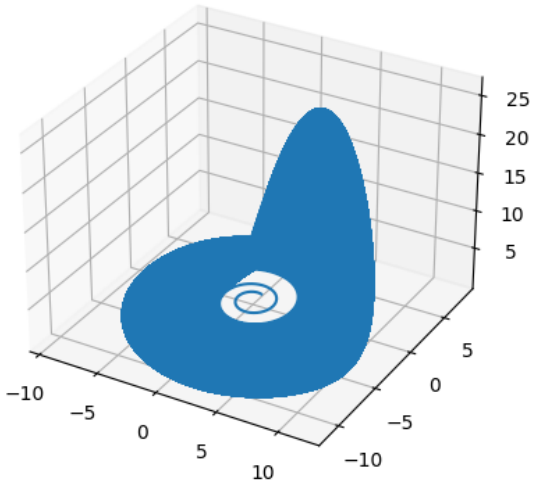
Decrypted Image

3.2. Rossler Model Example Image

Parameters used: (0.2, 0.2, 5.7)
Key used: (0.1, 1.0, 0.01, 0.01)

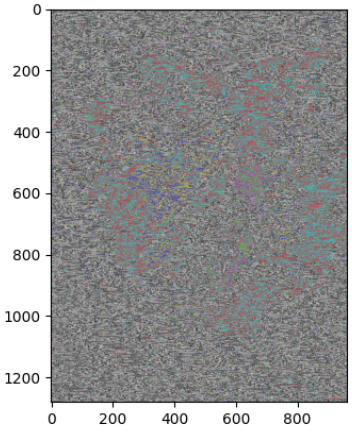
Obtained Images (Image captured by us):

Trajectory generated: Trajectory was generated for 960*1280 time steps of 0.01 dt hence it is filled.



Images obtained
Image size - 960 x 1280 pixels

Original Image

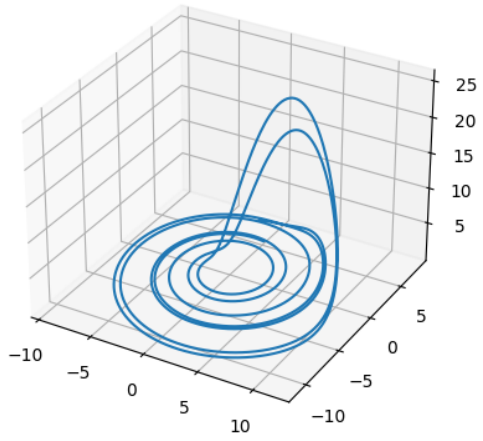


Encrypted Image

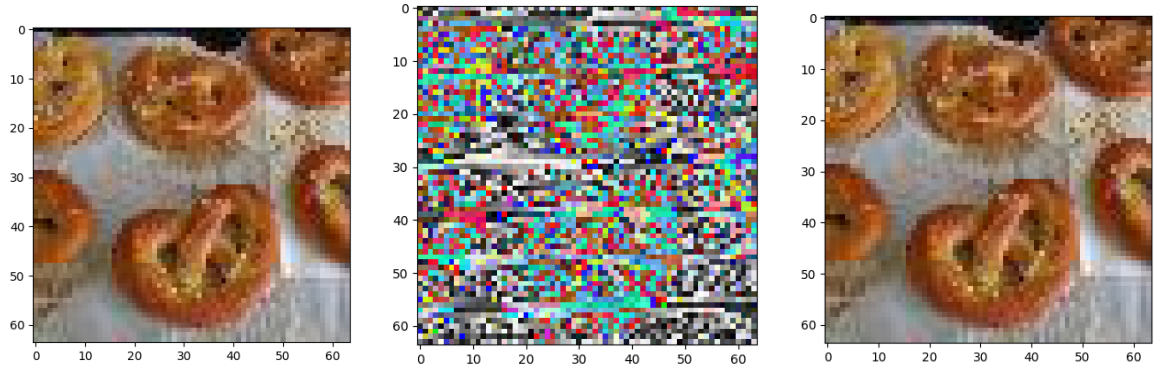
Decrypted Image

Obtained Images (from standard dataset - tiny-imagenet-200):

Trajectory generated:



Images obtained:
Image Size - 64 x 64 pixels



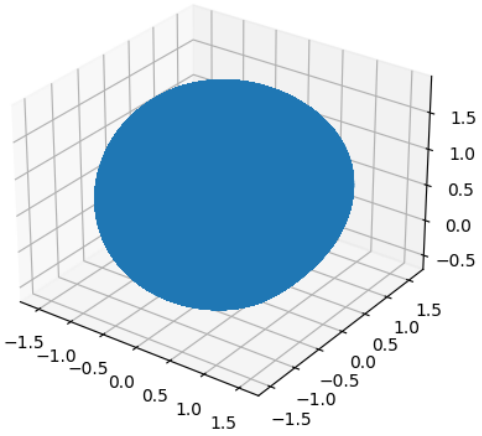
Original Image | Encrypted Image | Decrypted Image

3.3. Langford Model Example Image

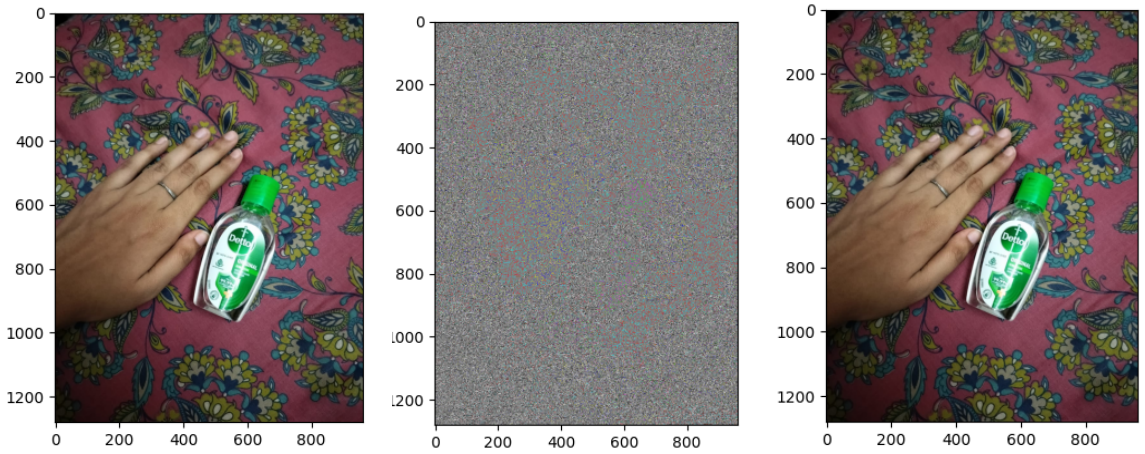
Parameters used: (0.95, 0.7, 0.6, 3.5, 0.25, 0.1)
Key used: (0.1, 1.0, 0.01, 0.01)

Obtained Images (Image captured by us):

Trajectory generated: Trajectory was generated for 960*1280 time steps of 0.01 dt hence it is filled.



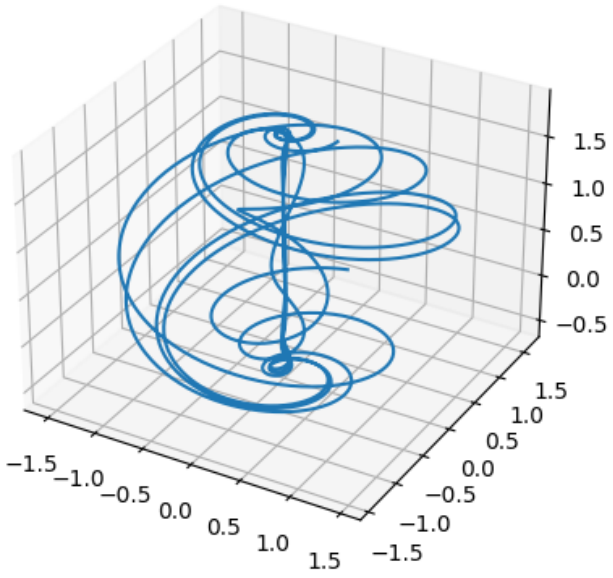
Images obtained
Image size - 960 x 1280 pixels



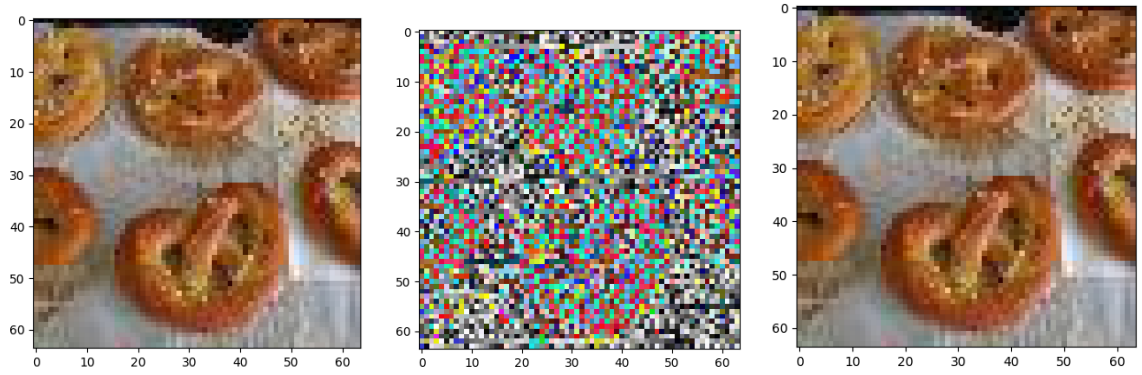
Original Image | Encrypted Image | Decrypted Image

Obtained Images (from standard dataset - tiny-imagenet-200):

Trajectory generated:



Obtained Images:



Original Image | Encrypted Image | Decrypted Image

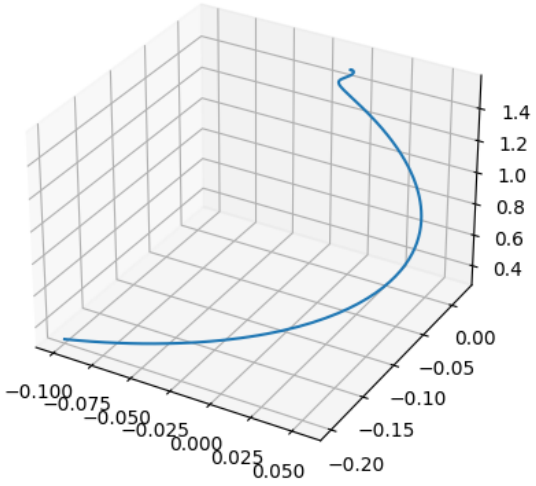
3.4. Non-Chaotic Example Images

The Langford model was initialised with parameters and initial points for which it does not show chaos. We can observe the effects of this below.

Parameters used: (0.95, 0.7, 0.6, 3.5, 0.25, 0.1)
Key used: (0.1, 1.0, 0.01, 0.01)

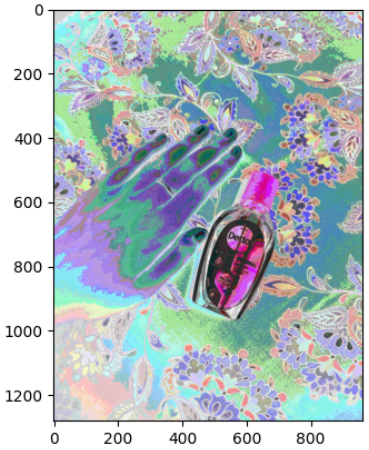
Obtained Images (Image captured by us):

Trajectory generated: Trajectory was generated for 960*1280 time steps of 0.01 dt hence it is filled.



Images obtained
Image size - 960 x 1280 pixels

Original Image

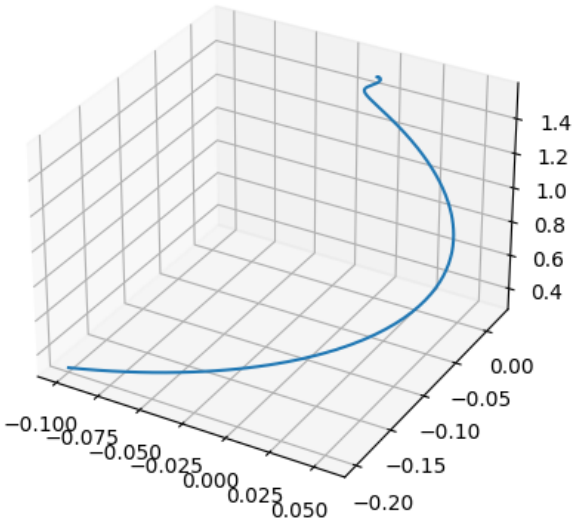


Encrypted Image

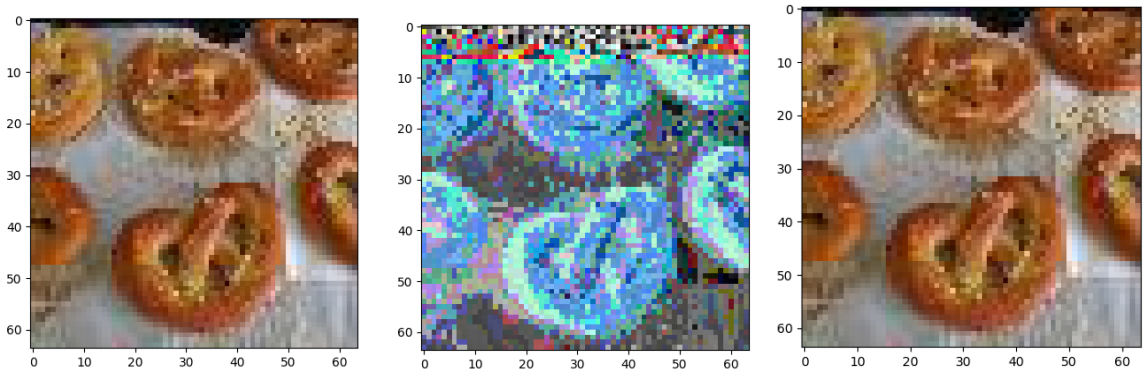
Decrypted Image

Obtained Images (from standard dataset - tiny-imagenet-200):

Trajectory generated:



Obtained Images:



Original Image | Encrypted Image | Decrypted Image

4. Assessment of the Quality of the Encryption

4.1. Structural Similarity Index

Several metrics can be used to compare the encryption efficiency of different algorithms. In this project the SSIM metric has been used and compared over different models.

SSIM (Structural Similarity Index) measures the similarity between two images, considering their luminance, contrast, and structure. It quantifies how much the structure of the encrypted image resembles the original image. The SSIM formula is:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Where:

- x and y are the original and encrypted images, respectively.
- μ_x and μ_y are the means of x and y.
- σ_x^2 and σ_y^2 are the variances of x and y.
- σ_{xy} is the covariance of x and y.
- c_1 and c_2 are constants to avoid instability near zero.

4.2. Results

The mean SSIM value and it's variance was calculated for each encryption system using models (Lorenz, Rossler and Langford systems) for 1000 images of size 64 x 64 pixels from the Tiny Imagenet 200 Dataset. A value of SSIM closer to 0 represents least similarity between the original and encrypted image, therefore it represents better encryption quality.

Results

Average SSIM for Rossler's Chaos: 0.0076303762957762224
Variance of SSIM for Rossler's Chaos: 0.0002378923309963854
Average SSIM for Lorenz's Chaos: 0.007655236729154438
Variance of SSIM for Lorenz's Chaos: 7.918936419754943e-05
Average SSIM for Langford's Chaos: 0.08365762628130442
Variance of SSIM for Langfords's Chaos: 0.0053506744131071125

Inference

Lorenz's system provides the most secure encryption, followed closely by Rossler's system. The encrypted images by Langford system are not very secure to be shared on open channels. A possible reason for this could be that Lorenz and Rossler systems show chaos for a wide range of starting points, therefore they are secure over several predefined keys (initial states). However, the Langford system only exhibits chaos for a smaller range of starting states.

5. Phase Portraits

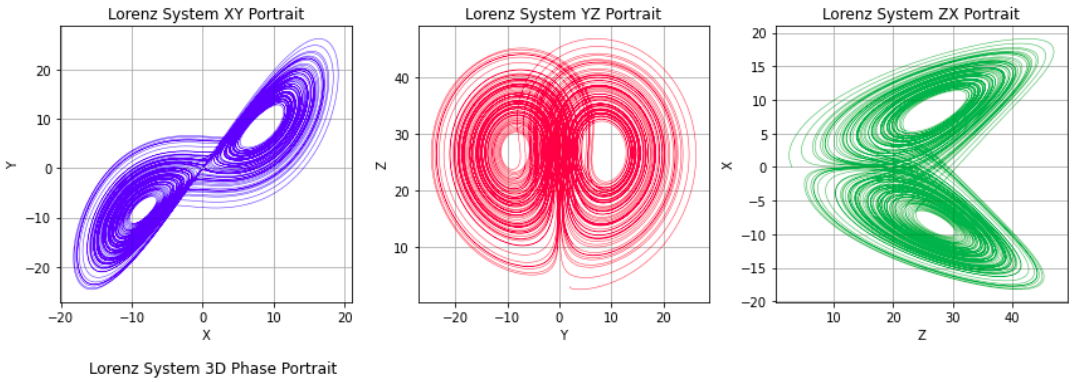
The phase portraits using the RK4 method in python have been added below. These portraits were used to study the systems in chaos before applying in image encryption.

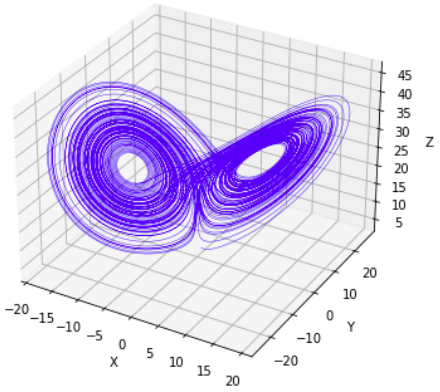
5.1. Lorenz System

Parameters

sigma = 10, rho = 28, beta = 8 / 3

Initial point - [0, 2, 3] , Step size - 0.0001 , Time - (0, 100)



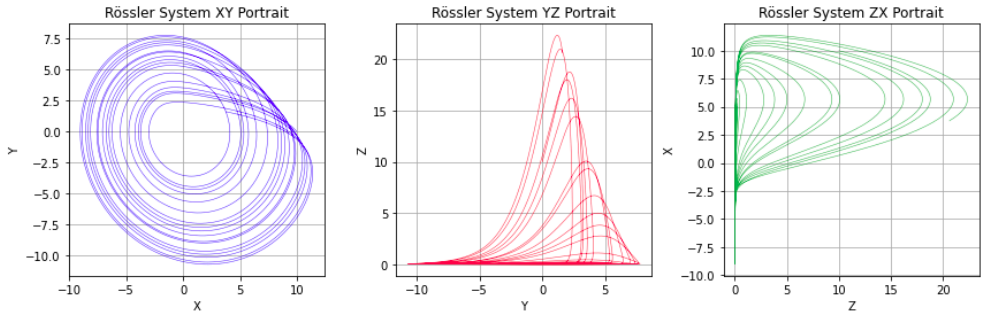


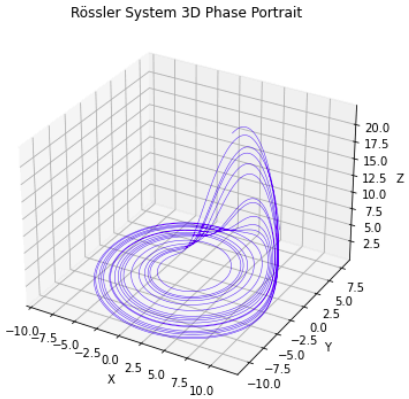
5.2. Rossler System

Parameters

a = 0.2, b = 0.2, c = 5.7

Initial point - [10, 0, 10] , Step size - 0.0001 , Time - (0, 100)



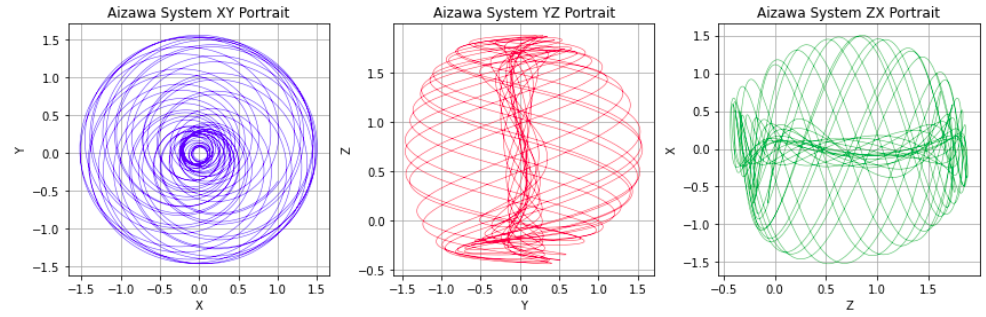


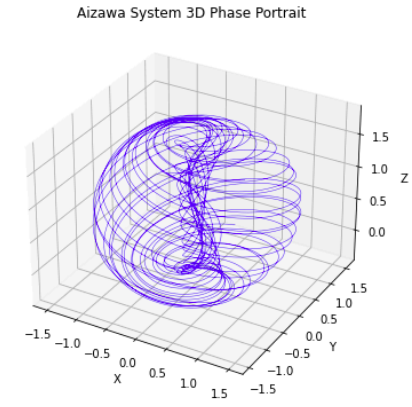
5.3. Rossler System

Parameters

a = 0.95, b = 0.7, c = 0.6, d = 3.5, e = 0.25, f = 0.1

Initial point - [0.1, 1.0, 0.01] , Step size - 0.0001 , Time - (0, 100)





6. Future Work

1. **Security Analysis**: Conduct a thorough security analysis of the encryption scheme by evaluating its resistance against common cryptographic attacks such as brute-force attacks, differential attacks, and statistical attacks. Compare it with existing cryptographic algorithms.
2. **Decryption Resolution**: Explore methods to improve the decryption resolution of the scheme, particularly in scenarios where noise or distortion may affect the encrypted image during transmission or storage. Investigate error correction techniques or adaptive algorithms to enhance the fidelity of decrypted images.
3. **Chaotic System Variants**: Experiment with a wider range of chaotic systems beyond the systems compared. Consider systems with different dynamic properties, attractor shapes, and bifurcation behaviours. Evaluate how the choice of chaotic system affects the encryption performance and robustness of your scheme.
4. **Physical Significance**: Delve deeper into the physical significance behind the differences observed in the results obtained from different chaotic models. Explore how the underlying dynamics of chaotic systems manifest in the encryption process and analyse the implications for security, efficiency, and computational complexity.
5. **Parameter Sensitivity Analysis**: Perform a sensitivity analysis to investigate how variations in the parameters of the chaotic systems impact the encryption quality and security of the scheme. Identify optimal parameter settings that maximise both security and performance metrics.

7. References

1. Introduction to Strange Attractors (https://live.ocw.mit.edu/courses/12-006j-nonlinear-dynamics-chaos-fall-2022/mit12_006jf22_lec19.pdf)
2. Lazaros Moysis (2024). Bifurcation diagram for the Rossler Chaotic system (https://www.mathworks.com/matlabcentral/fileexchange/157411-bifurcation-diagram-for-the-rossler-chaotic-system), MATLAB Central File Exchange. Retrieved April 12, 2024.
3. https://www.cfm.brown.edu/people/dobrush/
4. https://www.dynamicmath.xyz/strange-attractors/
5. https://github.com/Saransh-cpp/Chaotic-Encryption/
6. https://paperswithcode.com/dataset/tiny-imagenet